IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Flight Fare Prediction System Using Machine

[1]K. Tulasi Krishna Kumar, [2]G.Bhavan Kiran

[1] Associate Professor & Training & Placement Officer, [2] MCA Final Semester, Master of Computer Applications,

Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India

Abstract: A Flight Fare Prediction System using Machine Learning is an intelligent application developed to estimate airline ticket prices based on various travel-related factors such as departure date, source city, destination city, airline company, journey duration, travel class, and number of stops. Airline ticket prices are highly dynamic and fluctuate frequently due to multiple factors including customer demand, seasonal trends, holiday schedules, fuel prices, seat availability, and airline competition. Because of these continuously changing conditions, predicting flight fares manually becomes difficult and time-consuming for travelers. The proposed system aims to solve this problem by using Machine Learning techniques to analyze historical flight data and generate accurate fare predictions. The system begins with the collection of historical airline datasets containing details about previous flight bookings and ticket prices. The collected data undergoes preprocessing to improve data quality and prepare it for machine learning operations. During preprocessing, missing values are identified and handled appropriately, duplicate records are removed, and irrelevant attributes are filtered out. Since machine learning models require numerical input, categorical features such as airline names, source locations, destination cities, and travel classes are converted into machine-readable form using encoding techniques like one-hot encoding and label encoding. Numerical features such as duration and days left before departure may also be normalized or scaled to improve prediction performance. After preprocessing, different Machine Learning algorithms are applied to train the prediction model. Algorithms such as Linear Regression, Decision Tree Regression, and Random Forest Regression are commonly used because they effectively identify relationships between travel features and airfare prices. Linear Regression helps analyze the linear relationship between variables, Decision Tree handles non-linear data patterns through rule-based structures, and Random Forest improves prediction accuracy by combining multiple decision trees. The performance of these models is evaluated using statistical metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared Score. These evaluation techniques help determine the accuracy and efficiency of the prediction models.

Keywords: Flight Fare Prediction, Machine Learning, Data Pre-processing, Feature Engineering, Regression Models, Random Forest, Travel Analytics, Price Forecasting, Flask Web Application, Airline Ticket Pricing, Predictive Modelling, Python, Model Deployment, User Interface, Travel Planning.

I. Introduction

A Flight Fare Prediction System using Machine Learning is developed to estimate airline ticket prices using historical and real-time travel data. Flight prices vary frequently due to demand, season, route popularity, airline competition, and special events, making prediction difficult. [2]The system analyses factors such as source, destination, travel date, airline, duration, stops, and booking time to forecast ticket prices. Data pre-processing techniques like cleaning, handling missing values, and feature engineering improve model accuracy. Machine learning algorithms such as Linear Regression, Random Forest, and Gradient Boosting are used for prediction. The model is evaluated using metrics like MAE, RMSE, and R^2 score to ensure performance.

Advanced techniques such as time-series analysis and hyper parameter tuning further improve prediction accuracy. A recommendation feature can suggest the best time to book tickets for lower fares. The system can integrate real-time airline and travel website data for updated predictions. A user-friendly interface allows travellers to enter details and get fare predictions instantly. Cloud deployment improves accessibility, while security measures protect user data. Overall, this system helps travellers save money and demonstrates the practical use of AI in the travel industry.

II. Literature Survey

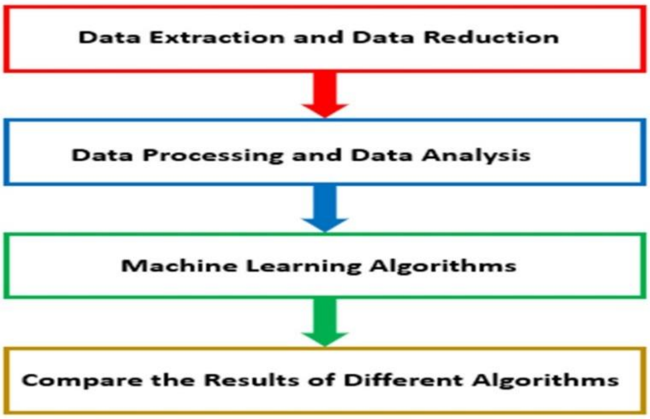
The prediction of flight ticket prices has become an important research area due to the rapid growth of the airline industry and the increasing demand for cost-effective travel. [11] Researchers have explored various Machine Learning and Data Mining techniques to analyse airline pricing strategies and forecast airfare trends. Early studies focused on statistical methods such as regression analysis to understand the relationship between travel factors and ticket prices [1]. These approaches showed that airfare depends on server including booking time, route distance, season, airline competition, and travel demand. However, traditional statistical models struggled to handle large and complex dataset with the advancement of Machine Learning, researchers began applying algorithms such as Linear Regression, Decision Trees, Random Forest, Support Vector Machines, and Gradient Boosting for airfare prediction. These models proved to be more accurate because they can learn complex patterns from historical data. Studies revealed that ensemble models like Random Forest and Gradient Boosting often outperform basic regression techniques. [9]

III. Challenges

Flight fare prediction is a complex task because airline ticket prices are highly dynamic and influenced by many unpredictable factors. [11] One of the biggest challenges is dynamic pricing, where ticket costs change frequently based on demand, season, seat availability, and competition among airlines. Collecting large and reliable historical flight data is also difficult because many airlines do not provide open access to their datasets, and web scraping may have technical and legal limitations. Even when data is available, it often contains missing values, duplicate records, and inconsistent formats, making data cleaning and pre-processing a time-consuming process. [16] Another major challenge is identifying the most relevant features that affect ticket prices, since too many irrelevant features can reduce model performance. Seasonal trends such as weekends, holidays, and festivals create additional variations that must be captured carefully. [1]There is also the risk of over fitting, where a model performs well on training data but fails to generalize to new data. Providing real-time predictions with high accuracy requires efficient system design and fast processing. [17] Airline pricing strategies change frequently, so models must be retrained regularly to remain accurate. Finally, ensuring data security and protecting user information is essential. [13]These challenges highlight the complexity of developing a reliable flight fare prediction system using machine learning.

IV. Proposed Methodology

The proposed methodology for the Flight Fare Prediction System follows a structured machine learning workflow to accurately estimate airline ticket prices using historical flight data. [6]The process begins with data collection, where flight information is gathered from airline websites, travel portals, and publicly available datasets. The dataset includes features such as airline name, source and destination cities, journey date, departure and arrival time, duration, number of stops, and ticket prices. After collecting the data, pre-processing is performed to clean the dataset by removing duplicates, handling missing values, and converting date and time into useful numerical formats.



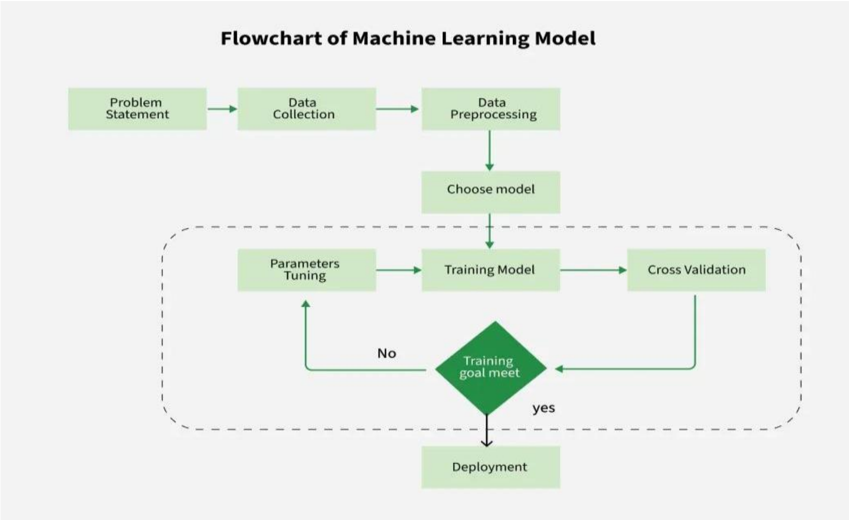
I. Architecture of Flight Fare Prediction System

Categorical variables like airline names and cities are encoded into numerical form so they can be used by machine learning algorithms. [3] Feature engineering is then applied to create meaningful features such as journey day, month, weekday, total travel duration, and indicators for weekend or peak season travel. [5] The prepared dataset is split into training and testing sets, and multiple regression algorithms such as Linear Regression, Decision Tree, Random Forest, and Gradient Boosting are trained to learn the relationship between travel details and ticket prices. [7]The models are evaluated using performance metrics like Mean Absolute Error, Root Mean Squared Error, and R^2 score to select the best-performing model. Finally, the chosen model is deployed as a web application or API, allowing users to enter travel details and receive instant fare predictions. [4] This methodology ensures an accurate, efficient, and user-friendly flight fare prediction system

The Flight Fare Prediction System follows a step-by-step process for accurate fare prediction. First, data extraction and reduction are performed to collect and filter relevant flight data. [8] Next, data processing and analysis are applied to clean and understand the dataset. Machine learning algorithms are then used to train models and predict ticket prices. [3] Finally, the results of different algorithms are compared to identify the best-performing model for accurate fare prediction. The Machine Learning Model used in the Flight Fare Prediction System follows a systematic and structured process to build an accurate and reliable airfare prediction application. The process begins with defining the problem statement, where the primary objective is to predict airline ticket prices based on travel-related factors such as airline name, source city, destination city, and departure date, duration of journey, travel class, and number of stops. Since flight ticket prices fluctuate frequently due to demand, seasonal variations, holidays, fuel costs, and seat availability, predicting fares accurately becomes a challenging task. [8] The project aims to solve this problem using Machine Learning techniques and predictive analytics. The next step involves collecting historical flight data from available datasets containing details about previous airline bookings and ticket prices. The dataset includes important attributes such as Airline Company, departure and arrival locations, total stops, journey duration, date of travel, and fare price. After data collection, preprocessing techniques are applied to improve the quality and usability of the dataset. During preprocessing, missing values are identified and handled, duplicate records are removed, and irrelevant data fields are eliminated. Since machine learning models work efficiently with numerical data, categorical features such as airline names, source cities, and destinations are converted into machine-readable form using encoding techniques like Label Encoding and One-Hot Encoding. Numerical attributes may also be scaled or normalized to improve model performance. After preprocessing, the dataset is divided into training and testing sets. Various Machine Learning algorithms such as Linear Regression, Decision Tree Regression, and Random Forest Regression are selected and trained using the prepared dataset. [17]

Linear Regression helps identify linear relationships between input variables and airfare prices, while Decision Tree and Random Forest algorithms capture complex and non-linear pricing patterns. These models learn from historical data and establish relationships between flight attributes and ticket costs. To improve the prediction accuracy and reliability of the system, cross-validation and hyper parameter tuning techniques are performed. Cross-validation helps evaluate the model using different subsets of data to ensure that the model generalizes well for unseen data. Hyper parameter tuning optimizes important model settings such as tree depth, number of estimators, and learning parameters to achieve better performance. The trained models are evaluated using performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared Score. [11] These metrics help measure prediction accuracy and determine the most suitable model for deployment. Once the machine learning model achieves satisfactory performance, it is deployed into a real-world web application environment using Flask. The trained model is saved as a joblib file and integrated with the backend system. Users can access the web application through a browser and enter travel details such as source, destination, airline, class, duration, and stops. [6]

The backend preprocesses the user input and sends it to the deployed machine learning model for prediction. Finally, the system displays the estimated flight fare instantly on the user interface. [13] The Flight Fare Prediction System provides significant benefits by helping travelers make better booking decisions, save money, and avoid high-price periods. It also demonstrates the practical implementation of Machine Learning, Data Analytics, Data Preprocessing, and Web Development technologies in solving real-world airline pricing problems.



I. Flowchart of Model

V. Algorithms and Techniques

1. User Lane Algorithm

The workflow of the Flight Fare Prediction System begins with the user interacting with the web application through a browser. Initially, the user opens the flight fare prediction website, which serves as the entry point to the system. The application then provides options for user registration and authentication. The user either signs up by creating a new account or logs in using previously registered credentials. This authentication process ensures that only authorized users can access the prediction services, thereby improving system security and protecting user data.

After successful login, the user is redirected to the prediction dashboard where flight-related details are entered. The user provides necessary information such as airline name, source city, destination city, travel class, number of stops, journey duration, and days left before departure. These parameters are essential because flight prices vary significantly depending on these factors. Once all details are entered, the user submits the form to initiate the fare prediction process.

One of the major advantages of this module is its user-friendly design. The interface allows users to easily provide input through dropdown menus and interactive fields without requiring technical knowledge. The system also saves time because users no longer need to manually compare fares across multiple websites. Another important advantage is secure authentication, which prevents unauthorized access and ensures privacy of user information. Furthermore, the system improves travel planning by helping users estimate future airfare costs before booking tickets.

2. WebApp Lane Algorithm

The WebApp layer is developed using the Flask framework and acts as the communication bridge between the user interface and the backend processing system. When the user enters login credentials, the Flask application validates the username and password against the stored database records. A decision node checks whether the credentials are valid. If the entered credentials are correct, access is granted and the user proceeds to the prediction page. If the credentials are incorrect, the system displays an error message and terminates the process until valid credentials are entered.

After authentication, the WebApp collects all flight-related inputs submitted by the user. Flask handles routing, request management, form validation, and session handling to ensure smooth navigation between pages. Once the form is submitted, the application transfers the collected data to the backend module for preprocessing and prediction. After receiving the prediction result from the backend, the Flask application dynamically displays the estimated flight fare on the result page.

The WebApp layer provides several advantages to the system. Firstly, it enables real-time interaction between users and the machine learning model, allowing instant predictions. Secondly, Flask is lightweight and efficient, which improves the overall performance and responsiveness of the application. Another important advantage is scalability, as additional functionalities such as flight booking integration, live airline APIs, and recommendation systems can be added easily in the future. The WebApp also improves user experience by providing structured navigation and interactive interfaces.

3. Backend Lane Algorithm

The backend layer performs the core machine learning operations required for flight fare prediction. After receiving the input data from the WebApp layer, the backend begins preprocessing the data. Since machine learning models require numerical input, categorical values such as airline name, source, destination, and travel class are converted into machine-readable format using encoding techniques like one-hot encoding. Numerical features such as duration and days left are normalized or scaled when necessary to improve model efficiency and prediction accuracy.

After preprocessing, the backend loads the trained machine learning model stored in a joblib file. In this project, a Linear Regression model is used because it efficiently identifies relationships between independent variables and flight prices. The pre-trained model has already learned patterns from historical flight fare datasets during the training phase. The processed user input is then fed into the model, which performs mathematical computations to estimate the predicted airfare. Once the prediction is generated, the backend sends the result back to the WebApp layer for display.

The backend module offers numerous advantages. One major advantage is prediction accuracy, as machine learning algorithms analyze large datasets and identify hidden pricing patterns that are difficult to recognize manually. Another advantage is automation, because the entire prediction process—from preprocessing to prediction generation—is completed automatically without human intervention. The backend also supports efficient data handling by processing both numerical and categorical data effectively. In addition, the system reduces human error since predictions are generated using trained algorithms rather than manual calculations. The backend architecture is also flexible and can support advanced machine learning models such as Random Forest, XGBoost, or Neural Networks for improved performance in future enhancements.

4. Overall System Workflow Algorithm

The complete workflow of the Flight Fare Prediction System begins when the user accesses the web application and authenticates using valid credentials. After successful login, the user enters flight details including airline, source, destination, travel class, stops, duration, and days left before departure. The Flask-based WebApp collects these details and forwards them to the backend processing system.

The backend preprocesses the input data using encoding and scaling techniques to convert it into machine-readable format. Next, the pre-trained Linear Regression model is loaded using the joblib library. The processed input data is supplied to the machine learning model, which analyzes the input features and predicts the estimated flight fare based on learned historical pricing patterns. The generated prediction is then transmitted back to the Flask application, which displays the predicted flight price to the user in real time. Finally, the workflow terminates after successfully presenting the output.

This overall workflow provides several important advantages. The system offers fast and real-time fare prediction, enabling users to make quick travel decisions. It also supports cost optimization by helping travelers identify suitable booking times and avoid high-price periods. The project demonstrates efficient integration of machine learning with web technologies, making it a valuable educational and research-oriented application. Additionally, the system is scalable, secure, and capable of handling large amounts of flight data efficiently. By automating fare prediction, the application minimizes manual effort and improves the reliability of airfare estimation.

The flow diagram you provided represents a vertical end-to-end process of the Flight Price Prediction System using a Flask-based web application. Here's a clear and concise explanation of each step:

V. Architecture

The Feature Engineering Module is an important stage in the Flight Fare Prediction System that improves the accuracy of the machine learning model. [14]After data preprocessing, raw data is transformed into meaningful features to help the model understand fare patterns better. [15]For example, journey dates are divided into day, month, and weekday, while departure and arrival times help calculate travel duration. Features like number of stops, peak season travel, and holidays are also included to reflect real-world pricing behavior. Encoding techniques such as One-Hot Encoding and Label Encoding convert categorical data into numerical form. Feature scaling may also be applied to improve model performance. By creating relevant and informative features, this module enhances learning and helps the system predict flight fares more accurately.

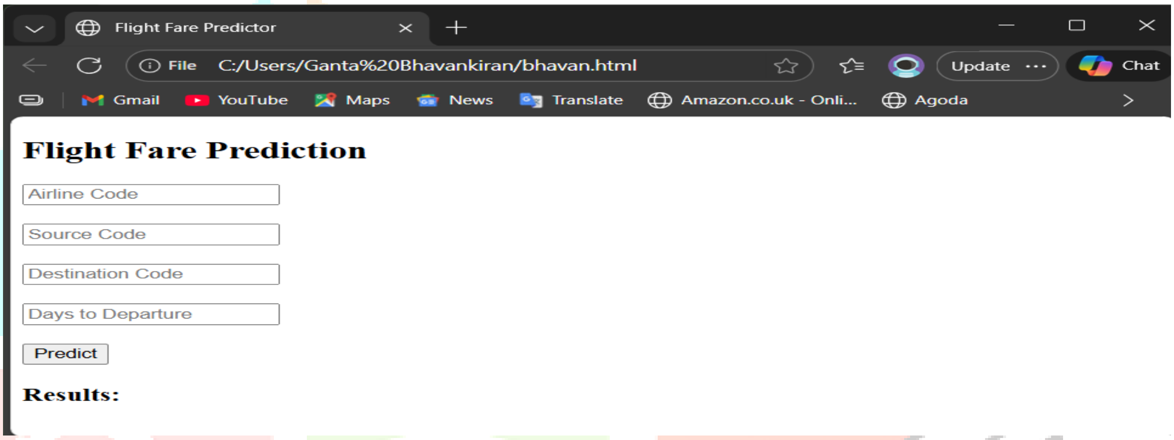
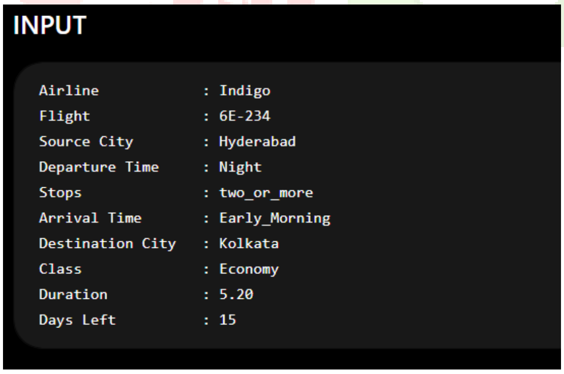
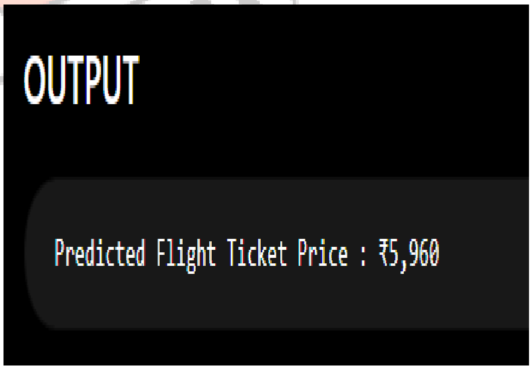


Figure 1a: Output screenshot of the Flight fare prediction

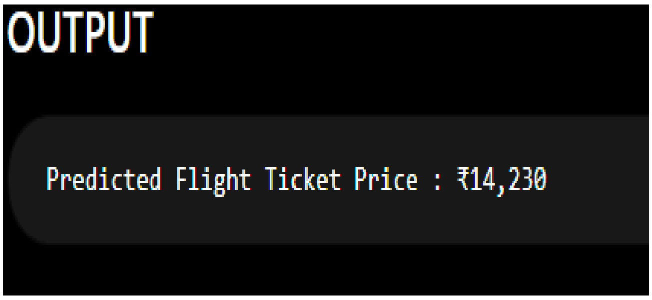




1b. Input **1c.Output**

The model provides four separate fare predictions corresponding to Economy, Premium Economy, Business, and First Class travel categories. This multi-output approach enables users to quickly understand price variations across classes and plan their journey more effectively.

2a.Customer Input



2b.Project Output

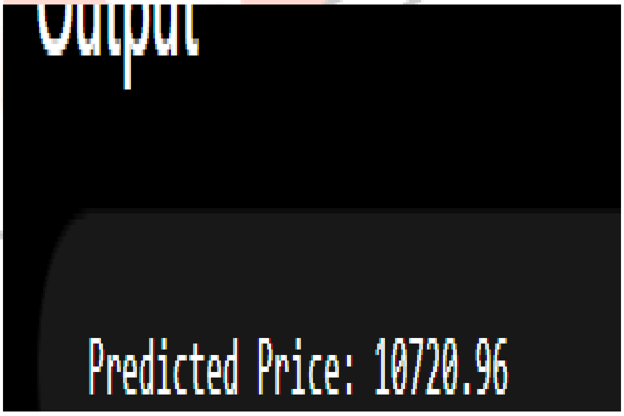
The model provides four separate fare predictions corresponding to Economy, Premium Economy, Business, and First Class travel categories. This multi-output approach enables users to quickly understand price variations across classes and plan their journey more effectively.

3aCustomer Input

3b.Project Output

The system generates four outputs representing predicted ticket prices for different travel classes: Economy, Premium Economy, Business, and First Class. These outputs help users compare fare options and choose the most suitable travel class based on their budget and preferences.

4a.Customer Input



4b.Project Output

The model provides four separate fare predictions corresponding to Economy, Premium Economy, Business, and First Class travel categories. [18]This multi-output approach enables users to quickly understand price variations across classes and plan their journey more effectively

VII. Conclusion

In conclusion, the Employee Burnout Detection System provides an intelligent solution for monitoring employee well-being using Machine Learning and Natural Language Processing (NLP) techniques. The system analyzes employee work patterns, behavioral data, and textual feedback to predict burnout levels accurately. By using the Random Forest algorithm, sentiment analysis, and chatbot interaction, the application helps identify stress and burnout risks at an early stage. The interactive dashboard and graphical visualizations make the prediction results easy to understand and monitor. Overall, the project supports organizations in improving workplace productivity, employee health, and decision-making through data-driven insights.

VII. Future Scope

In the future, the Employee Burnout Detection System can be enhanced by integrating advanced Machine Learning and Deep Learning algorithms to improve prediction accuracy. The system can be expanded with real-time monitoring, wearable device integration, and mobile application support for continuous employee wellness tracking. Advanced NLP techniques such as multilingual sentiment analysis and voice emotion recognition can further improve chatbot interactions. Integration with HR management systems and cloud platforms can help organizations manage employee performance, stress levels, and workplace productivity more effectively. Overall, the project has strong potential to evolve into a smart AI-driven employee wellness and mental health management platform.

References

1. Brownlee, J. (2018). Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End. Machine Learning Mastery.
2. Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.
3. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
5. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.
6. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques (4th ed.). Morgan Kaufmann.
7. Aurélien, G. (2019). Practical Machine Learning Projects with Python. Packt Publishing.
8. Zhang, Y., & Wu, L. (2020). "Airline Ticket Price Prediction Using Machine Learning Techniques," International Journal of Computer Applications, Vol. 175, No. 32.
9. Kumar, R., & Singh, P. (2021). "Comparative Analysis of Regression Models for Flight Fare Prediction," Journal of Artificial Intelligence and Data Science, Vol. 5, No. 2.
10. Lee, H., & Park, S. (2020). "Machine Learning Approaches for Dynamic Airline Pricing," Proceedings of the International Conference on Big Data Analytics and Computing.
11. TensorFlow Documentation (2023). An End-to-End Open Source Machine Learning Platform. Retrieved from: https://www.tensorflow.org/
12. Keras Documentation (2023). Deep Learning for Humans. Retrieved from: https://keras.io/
13. NumPy Documentation (2023). Fundamental Package for Scientific Computing with Python. Retrieved from: https://numpy.org/
14. Matplotlib Documentation (2023). Visualization with Python. Retrieved from: https://matplotlib.org/
15. Seaborn Documentation (2023). Statistical Data Visualization. Retrieved from: https://seaborn.pydata.org/
16. SQLite Documentation (2023). Self-Contained SQL Database Engine. Retrieved from: https://www.sqlite.org/
17. Apache Software Foundation (2023). Apache Spark: Unified Analytics Engine for Big Data. Retrieved from: https://spark.apache.org/
18. IBM Cloud Education (2022). "Predictive Analytics Using Machine Learning for Transportation Systems," IBM Research Publications.
19. Ng, A. (2019). Machine Learning Yearning: Technical Strategy for AI Engineers. DeepLearning.AI.
20. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, Vol. 12, pp. 2825–2830.

BIBLOGRAPHY

K. Tulasi Krishna Kumar is a Ratified Associate Professor affiliated with Andhra University and currently serves as the Placement Officer at SVPEC. With over 17 years of distinguished experience, he has played a pivotal role in training and placing students across IT, ITES, and core industry sectors, having mentored more than 16,000 students and 750 faculty members. An accomplished academician, authored eight books and successfully guided over 85 undergraduate and postgraduate project teams. He has also contributed extensively to research, with guidance leading to more than 75 publications in reputed international journals. A Certified Campus Recruitment Trainer (JNTUA), he holds an M.Tech in Computer Science and is presently pursuing his Ph.D. in CSE.

G.Bhavan kiran is currently pursuing his final semester of Master of Computer Applications (MCA) at Sanketika Vidya Parishad Engineering College, which is accredited with an ‘A’ grade by NAAC, affiliated to Andhra University, and approved by AICTE. With a keen interest in Machine Learning and Artificial Intelligence, she has undertaken her postgraduate project titled “Idea Crafter: An Intelligent ML-Powered Chat Bot for Personalized Entrepreneurial Mentorship and Startup Guidance ” The project is carried out under the guidance of K.T. Krishna Kumar, assistant professor and training & placement officer, svpec